Heart Failure Prediction

109061521 林依蓁 109062702 楊晴雯

一、動機與目的

心血管疾病是全球第一大死因，每年帶走1790萬條性命，而其中最常引起的事件又是心臟衰竭，因此**我們希望幫助醫院訓練一個model，能預測出可能因為心臟衰竭而死亡的高風險族群**， 使醫生能針對這些病患做進一步的健康評估、使病患及早接受治療，也提醒高風險病患本身該注意生活作息，飲食起居等等，降低因心臟衰竭而死亡的機率。

二、Dataset介紹

我們的dataset (Fig 1.)取自於kaggle，當中蒐集了**299位病患**的資料，包含年齡、貧血、肌酸磷化酶(CPK)、糖尿病、射血分數、高血壓、血清肌酸酐、血清鈉、性別、抽菸與否、期間共**12種feature**。其中training data跟 testing data的比率分成8 : 2。

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
0	75.0	0	582	0	20	1	265000.00	1.9	130	1	0	4	1
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1	0	6	1
2	65.0	0	146	0	20	0	162000.00	1.3	129	1	1	7	1
3	50.0	1	111	0	20	0	210000.00	1.9	137	1	0	7	1
4	65.0	1	160	1	20	0	327000.00	2.7	116	0	0	8	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
294	62.0	0	61	1	38	1	155000.00	1.1	143	1	1	270	0
295	55.0	0	1820	0	38	0	270000.00	1.2	139	0	0	271	0
296	45.0	0	2060	1	60	0	742000.00	0.8	138	0	0	278	0
297	45.0	0	2413	0	38	0	140000.00	1.4	140	1	1	280	0
298	50.0	0	196	0	45	0	395000.00	1.6	136	1	1	285	0

[299 rows x 13 columns]

Fig 1. Heart Failure Dataset

三、方法

Fig 2是我們的flow chart，一開始將dataset當作input傳入，在feature preprocessing中，將data進行標準化、透過feature selection選出當中較重要的幾項feature，以及利用PCA做降維的動作，再來對多種classifier個別做訓練，比較他們evaluation後的結果，最後再透過Tune參數達到更好的performance。

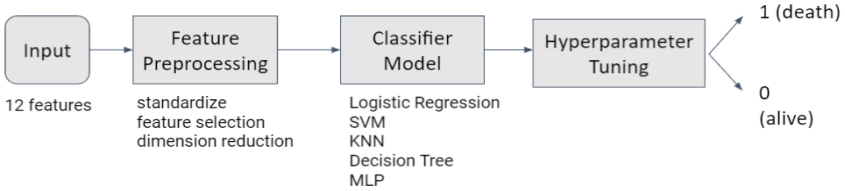

Fig 2. Flow Chart

- **Feature Preprocessing**

  1. Standardize (採用standardscaler)

  從Fig 3.可以觀察到，datasets 中feature間的scaler差異非常的大，最小的只有零點多，最大的卻有到八十幾萬，因此需要對data做標準化來排除不同數據間極大的落差，有助於後續的分析及比較。至於標準化的部分我所採用的方式是 standardscaler 將原始data轉成normal distribution ，也就是mean調成0，variance調成1

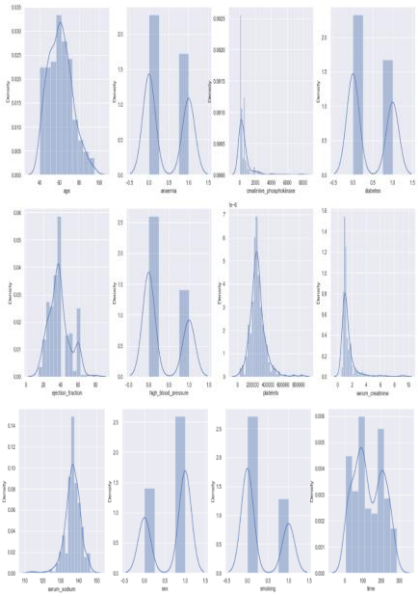

Fig 3. Feature Scale

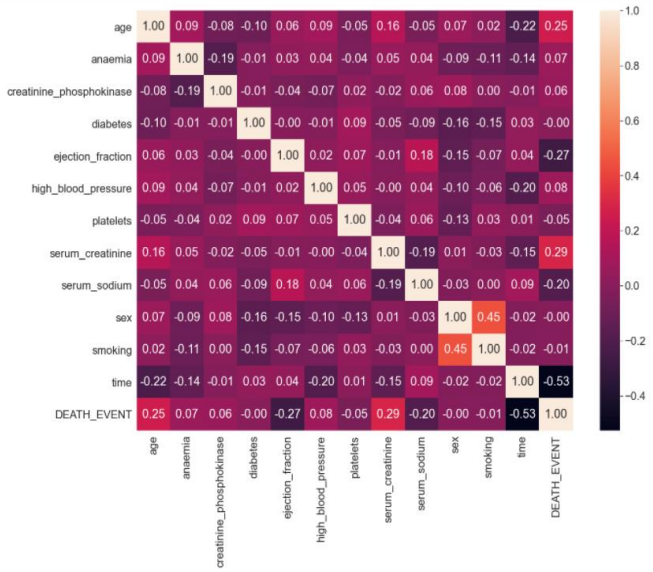

Fig 4. HeatMap

2. Feature Selection

a. Filter Method

在Filter Method中選擇features時，只考慮features與target間的相關性，進而對features進行評分，可以選擇固定的feature個數或者設定一個threshold並選取threshold內的features對我們的model進行訓練。

我們利用correlation來選出features，Fig 4表示的是各features與target間的correlation，我們取出相關性較高的前3項 (time, ejection_fraction, serum_creatinine)和前5項 (time, ejection_fraction, serum_creatinine, age, serum_sodium)的features來做分析。

b. Wrapper Method

Wrapper method是先選定一個base model ，然後根據該model去選擇適當的feature組合，因此每個model中採用的feature數量及組合會不相同，它的流程可以看到Fig 5，一開始先用所有的feature去train model，找到最佳的model參數解，再透過不同feature的組合 (ex. 在每次做計算時選擇若干或排除若干個feature)，最後performance最高的feature組合即為最佳的feature數量，也因為model的參數是透過all feature所train出來的，因此這個方法的performance基本上會優於或等於all feature的performance。

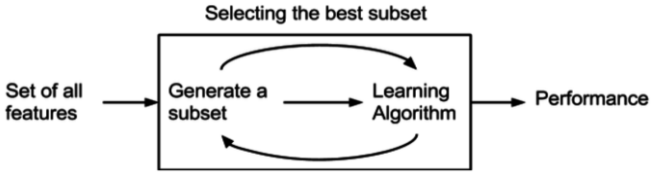

Fig 5. Wrapper Method

3. Feature Reduction

a. PCA

PCA的目標是通過某種線性投影，將高維的資料對映到低維的空間中表示，即把原先的n個特徵用數目更少的m個特徵取代，新特徵是舊特徵的線性組合。當我們將features利用PCA降至2維以及3維時(如Fig 6)，發現無法準確的將data區分開來，因此我們又畫出Fig 7去查看各component的所佔的variance，並發現我們必須取到9個components才能將原始資料80%的information呈現出來。

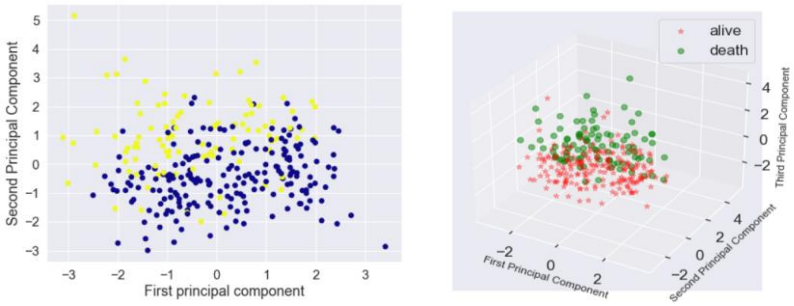

Fig 6. Visualization of PCA

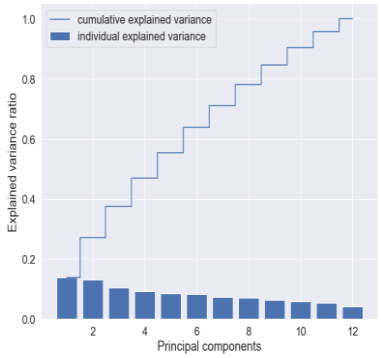

Fig 7. Variance Ratio of Principal Components

● Classifier Model

1. Logistic Regression

用線性回歸的輸出來判斷這個資料屬不屬於target，相當簡單的概念，將點帶進去回歸線，回歸線輸出值若是>=0，是一類(target)，值<0是另一類(non-target)。

Pros	Cons
● 分類時計算量非常小，速度很快 ● 計算代價不高，易於理解和實現	● 當特徵空間很大時，性能不是很好 ● 容易欠擬合，一般準確度不太高 ● 只能處理兩分類問題，且必須線性可分，對於非線性特徵，需要進行轉換

2. SVM

是一種supervised的學習方法，用統計風險最小化的原則來估計一個分類的超平面(hyperplane)，其基礎的概念非常簡單，就是找到一個決策邊界(decision boundary)讓兩類之間的邊界(margins)最大化，使其可以完美區隔開來。

Pros	Cons
● 可以解決高維問題，即大型特徵空間 ● 能夠處理非線性特徵的相互作用 ● 泛化能力比較強	● 當觀測樣本很多時，效率並不是很高 ● 對非線性問題沒有通用解決方案，有時候很難找到一個合適的核函數 ● 常規SVM只支持二分類

3. KNN

(K-NearestNeighbor)該演算法的核心思想是找到距離最近的K個鄰居→進行投票→決定類別

Pros	Cons
● 理論簡單，容易實現 ● 可做classification也可做regression ● 準確度高，對outlier不敏感	● 計算量大，dataset太大時不適合使用 ● 維度太高時，資料處理效果不好 ● 樣本不平衡時，預測樣本數較少的類別Accuracy會較低 ● 預測速度比logistic regression之類的演算法緩慢 ● 解釋性不如Decision Tree高 ● K值大小的選擇問題

4. Decision Tree

決策樹演算法採用樹形結構，使用層層推理來實現最終的分類。預測時，在樹的內部節點處用某一屬性值進行判斷，根據判斷結果決定進入哪個分支節點，直到到達葉節點處，得到分類結果。這是一種基於 if-then-else 規則的監督式學習算法，決策樹的這些規則是通過訓練得到，而不是人工制定的。

Pros	Cons
● 易於實現，可解釋性強	● 結果可能是不穩定的，因為在資料中一個很小的變化可能導致生成一個完全不同的樹，這個問題可以通過使用整合決策樹來解決

5. MLP

主要由多層神經元構成的神經網路組成，包括輸入層、中間層和輸出層，層與層之間是全連接的，除了輸入層，其他層每個神經元包含一個激活函數。

Pros	Cons
● 可處理非線性問題 ● 良好的容錯率	● hidden node個數選擇困難 ● 學習速度慢 ● 容易陷入區域極值

四、實驗結果

為了更準確評估預測的結果，我們採用accuracy, recall, precision以及f1-score這四項指標來評估model，所有的model都有測試過不同的參數去找出最佳的參數組合。

1. All Features

Table 1是採用 All features所得到的結果，可以發現model選用decision tree時的accuracy是最高的。

Table 1. Result of All Features

	LR	SVM	KNN	Decision Tree	MLP
Accuracy	90.00%	88.33%	80.00%	93.33%	88.33%
Recall	90.00%	88.33%	80.00%	93.33%	88.33%
Precision	90.68%	88.09%	77.98%	93.26%	88.09%
F1-score	90.22%	88.18%	77.88%	93.15%	88.18%

2. Filter Method

透過HeatMap分別選出5個features以及3個features，結果分別如Table 2及Table 3所示。

在**5個features**時表現最佳的是Decision Tree，且這裡的feature數量小於前面all features的12個features。

選**3個feature時**表現最佳的是Logistic regression，可發現Logistic Regression在feature數量較小時能有較好的表現。

Table 2. Result of Filter Method (5 Features)

	LR	SVM	KNN	Decision Tree	MLP
Accuracy	88.33%	88.33%	80.00%	93.33%	88.33%
Recall	88.33%	88.33%	80.00%	93.33%	88.33%
Precision	89.57%	88.09%	77.98%	93.26%	88.67%
F1-score	88.70%	88.18%	77.88%	93.15%	88.47%

Table 3. Result of Filter Method (3 Features)

	LR	SVM	KNN	Decision Tree	MLP

Accuracy	88.33%	86.67%	86.67%	88.33%	86.67%
Recall	88.33%	86.67%	86.67%	88.33%	86.67%
Precision	88.67%	86.67%	86.67%	87.93%	86.67%
F1-score	88.47%	86.67%	86.67%	87.82%	86.67%

3. Wrapper Method

model的參數和all feature train出來最好的結果所採用的參數是一樣的，只是feature數量相比下會減少，因此從Table 1以及Table 4這兩張表格發現，wrapper method的方法基本上會優於或等於all features 時的結果。在此方法下，decision tree所得到的結果也是最好的，但在此方法下decision tree只利用了10個features。其他model所採用的feature數量也在表格上呈現了。

Table 4. Result of Wrapper Method

	LR	SVM	KNN	Decision Tree	MLP
Accuracy	90%	91.67%	88.33%	93.33%	81.67%
Recall	90%	91.67%	88.33%	93.33%	81.67%
Precision	90.68%	92.48%	88.09%	93.26%	82.15%
F1-score	90.22%	90.98%	88.18%	93.15%	81.88%

4. PCA

在PCA中我們利用了9個components，在此方法中表現較佳的是logistic regression，它的accuracy, recall, precision f1-score皆達到93.33%的performance。

Table 5. Result of PCA

	LR	SVM	KNN	Decision Tree	MLP
Accuracy	93.33%	85.00%	80.00%	78.33%	86.67%
Recall	93.33%	85.00%	80.00%	78.33%	86.67%
Precision	93.33%	84.25%	78.02%	77.81%	86.13%
F1-score	93.33%	84.34%	76.80%	78.05%	85.83%

5. Other Tests

在上述的方法中，accuracy最高只能達到93.33%，因此我們想利用其他方法來進一步提升我們的accuracy。

a. Weighted Important Features - Copy Important Features

我們第一個想到的方法是加重重要feature的權重，利用HeatMap來選出5個最重要的features，而加重權重的部份則採用複製多次這些重要的features，進而讓這些features的權重比其他features的權重變更高。但採用此方式出來的結果並沒有更好(如Table 6所示)。

Table 6. Result of Weighted Important Features

	LR	SVM	KNN	Decision Tree	MLP
Accuracy	90%	90%	90%	91.67%	88.33%
Recall	90%	90%	90%	91.67%	88.33%
Precision	90%	90%	89.72%	91.52%	88.67%
F1-score	89.38%	89.38%	89.72%	91.56%	88. 47%

b. Outlier

第二個想到的方法是移除outliers，因為outliers會影響到model訓練的結果，可能導致accuracy降低，因此希望藉由移除outliers提升Accuracy。

不過在實驗過程中發現將outliers移除後的準確率，只剩六七十左右，我們認為可能原因是因為dataset筆數太少所造成，導致準確率不升反降。

c. Ensemble Learning

主要是經由結合多個機器學習器而成的大模型。透過不同方法綜合起來得到最終的結果。又可細分成下列三種方法。

(i.) Bagging

隨機抽取樣本及特徵，並以此內容物建模。接下來將樣本放回，再抽一次樣本及特徵建構出第二個小模型，後續小模型以此類推。最後將每個模型產出的結果等權重加權，決定最終的結果。

⇨ 在此方法中，**最佳的Accuracy為91.66%** (feature = 5下，使用Logistic Regression和10個小模型)

(ii.) Boosting

Boosting裡的分類器則會由前一個分類器的結果而做更進一步的修正，因此每個分類器皆有所關連，不同於Bagging使用相同權重，Boosting會給予準確度高的model較高的權重。

⇨ 在此方法中，**最佳的Accuracy為90.00%** (feature = 5下，使用Ada Boosting, base model 為Decision Tree)

(iii.) Voting

上述兩種方法都只能選一種學習器，沒辦法交錯使用，Voting則可以將”不同類型”的弱學習器結合在一起。

⇨ 在此方法中，**最佳的Accuracy為90.00%** (feature = 5下，使用Logistic Regression + SVM + KNN + Decision Tree + MLP)

五、結論

在現有的五種model下(Logistic Regression、SVM、KNN、Decision Tree以及MLP)，我們最佳的準確率為93.33%(Feature=5，使用Decision Tree 的Classifier)。為了使準確率能高於93.33%，我們仍嘗試其他多種方法，包含(1)提高important features的權重、(2)移除outliers以及(3)ensemble learning這三種方式，但是這些方法並無法超越現有最佳的準確率(93.33%)，甚至出現下降的現象。我們認為此問題可能的原因為我們所選用的dataset筆數過少(僅有299位病患)，才導致準確率最高只能到93.33%，若要讓準確率繼續提升，需要再收集更多的dataset才可能達到，不過醫學相關的資料礙於病人隱私權、需專業醫師協助標記label等原因，在蒐集上有相當的難度，因此目前無法達成。

六、Reference

dataset: https://www.kaggle.com/andrewmvd/heart-failure-clinical-data
ensemble method:
https://pyecontech.com/2020/11/30/python_ensemble_learning_bagging/
https://pyecontech.com/2020/12/18/python_ensemble_learning_boosting/
https://pyecontech.com/2020/12/27/python_ensemble_learning_voting/